What Is Natural ISPF?

Natural ISPF (Integrated Structured Programming Facility) is Software AG’s application development tool for the building, testing and maintenance of applications throughout their life cycle.

So what? you may ask. In answer, consider the following issues:

- *Does the building and testing of applications at your site take place on different operating systems and/or in different TP/DP environments using different interfaces and tools?*

  Natural ISPF is independent of your site’s operating system(s) and TP/DP environment(s). Users of Natural ISPF can stay in the same working environment, irrespective of the underlying operating environment and TP monitor currently used. CICS, IMS, TSO, UTM, TIAM, Batch are just some of the more common systems supported by Natural ISPF.

- *Would it help you if you had a single, familiar system image of all objects and resources involved in application development?*

  Using Natural ISPF, Natural and non-Natural objects can be accessed and processed with a single user interface. You can display and edit text files, JCL, and code held as Natural objects or 3GL programs (Assembler, COBOL, etc.). You can submit and monitor jobs, control job listings and perform dataset maintenance tasks. Operations on all of these different objects types, whether they reside on OS/390, VSE/ESA or BS2000/OSD, is afforded using standard menus and a single set of commands with consistent syntax.

- *How many different editors do your application programmers use in their development work?*

  Natural ISPF uses only one editor to list, display and edit all objects (files, members, JCL, job output, Natural programs, etc.). Object lists and the edit environment are presented in a way that is comfortable and largely self-explanatory, and looks especially familiar to programmers who are used to working in time-sharing environments (for example, TSO/ISPF on OS/390).

- *Are you having or anticipating resource bottlenecks in your TSO/ISPF environment?*

  Possible problems of overhead and resource shortages in growing TSO/ISPF environments is just one example of a burning data processing issue that Natural ISPF addresses. The introduction of Natural ISPF to your site allows easy migration of data processing methods to other environments, while preserving, even enhancing ISPF capabilities and functionality.

The above are just some of the issues Natural ISPF solves. In addition, Natural ISPF offers a wide range of additional capabilities and features (outlined in this document) that make it the perfect companion to Natural for sites at which application development takes place. The benefits to be reaped from introducing Natural ISPF to your data processing setup are many. Among the most obvious are:

- **Reduced training costs:**

  The independence from operating systems and TP environments provided by Natural ISPF means reduced investment in training otherwise required for the different environments. Only one working environment is used and one syntax needs to be learned.

- **Increased productivity:**

  Use of a single working environment, together with the flexible, advanced facilities offered by Natural ISPF such as multiple parallel sessions, cross-session operations, automatic generation of skeleton programs, customization of the users’ environment to their individual needs and ease of use substantially shortens the turnaround time in application development from specification to completion.

Copyright Software AG 2002

- **Resource savings:**

  With Natural ISPF offering capabilities familiar to TSO/ISPF users in other environments, the migration of data processing methods becomes a smooth operation that involves little or no retraining, yet results in substantial savings on computer resources.

- **Uniform interface for all users:**

  The scope of Natural ISPF functionality, the wide range of accessible objects, and the high level of integration potential with other products make Natural ISPF a powerful tool not only for application developers, but also for system programmers, computer operators and general users. Having a single, uniform view of your data processing resources makes work for these individual groups more convenient and makes communication between them easier and more effective, thus streamlining your whole operation.

Copyright Software AG 2002